

CS442 Fall 2008: Project Description

Instructor: Vinod Ganapathy
vinodg@cs.rutgers.edu

TA: Chih-Cheng Chang
geniusc@cs.rutgers.edu

Out: September 29, 2008.

The class project is worth 20% of your grade in the course. It is due at the end of the semester, when you will be expected to demonstrate a working system. The project involves a significant design and programming component and will require you to start work early and work regularly over the course of the semester.

You will work on this project in groups of three or four students. You may, however, discuss design and implementation details with students outside your group. However, you are expected to write up your own design documents and work on your own code. Copying code/design decisions from other students in class will be considered academic dishonesty and will be dealt with severely, according to University policy.

Project goal

The goal of this project is to build a privacy-aware social networking application using mobile phones. Each one of you will be provided with a mobile phone that is equipped with a Global Positioning System (GPS). Using GPS, your phone will be able to determine its precise geographic location (latitude/longitude). Working in teams of three/four, you will use this phone to build a social networking application. Your social network includes you and your project partners.

The social networking application that you will build will be structured as a client and a server, with the client running on your mobile phone and the server running on a department machine. You are required to build both the client and the server. The goal of this social networking application is to notify you of the presence of your social network peers in your geographic proximity. For example, your social network application must be able to tell you whether your project partner is also in the same physical location as you are, e.g., the Busch Campus Center. Additionally, this application must also privacy preserving, i.e., it will present the user with an option to protect his/her location information from peers in the user's social network. If this option is enabled, then the user will be able to update the server with information about their location, but social network peers will be unable to determine that the user is in their geographic proximity.

To implement this system, we recommend that you proceed as following. First, adapt the template code that you will be supplied with (a simple client/server calculator application) to work with the cell phones. That is, get the clients to run on the cell phones and run the server on a department machine. This process will acquaint you with the cell phone as a programming environment. Second, learn how to obtain your geographic coordinates by reading GPS location

information from the cell phone. Third, implement a basic social networking application (both the server and the client) that will allow the client to send periodic updates on the client's location to the server. When the client requests the server to send information on social networking peers in his/her geographic proximity, the server must identify other peers "close" to the user (you are free to come up with your own definition of "close") and send this list of peers back to the client. You should aim to have all of this working by mid-November. Finally, you should augment the client and the server to implement privacy protection. The client should be able to tell the server to hide his/her location information. The server in turn, will note this information, and will not return information about this client in query results for other social network peers.

Source code

The system that you will build is a Java-based client/server application, where the client runs on your OpenMoko cell-phone. In order to get you started on your implementation, we are providing some template code that you may find useful to begin work. We have provided you with a simple Java-based client/server *calculator* application. The client allows a user to enter simple expressions, such as "2 + 3." These expressions are transferred via a network connection to the server, where they are evaluated, and the result is sent back to the client where it is displayed back to the user. All network communication happens via XML messages and is secured using the Secure Sockets Layer (SSL). You can download this source code from the class webpage.

You can use this code as a starting point for your own implementation. You will need to appropriately modify the client, the server, and the format of the XML messages exchanged between the server and the client to enable communication.

1. You will have to modify the client application to be a social-networking client. Upon requesting a request from the user to synchronize his/her location with the social networking server, this client will have to query GPS-based location information provided by the OpenMoko cell phone and send it to the server. In turn, the server will respond with a list of friends in the same geographic area, which the client will display to the user. Similarly, the client must also present the user with an interface via which he can communicate his intent to the server to hide his presence.
2. You will have to modify the server application to first implement basic social networking, and then to implement access control. First, you will have to modify the server so that when it receives location information from a client (say, Alice), it searches its database and returns other clients who may be in the same geographic region as Alice. You will also have to update a database to store the current location of Alice. Second, you will have to implement access control on this server. Clients may request that their location information be hidden from their friends, and the server will have to suppress the location of such clients when it answers queries from other clients.
3. You will have to modify the XML messaging format by suitably modifying the XML Document Type Definition (DTD). Your messages will now be location updates from the client to the server and answers to location queries from the server to the client.

Project design document

Each team is expected to write up a detailed project design document. Your design document must describe in detail your proposal to implement the client and the server. That is,

1. describe in detail the list of features that will be presented to the user as part of the client application.
2. describe in detail the list of features that will be supported by the social networking server.
3. describe how the server and client would communicate with each other.
4. describe how you propose to send location updates from the client to the server (i.e., is it going to be periodic, or will the user have to do it manually, etc.).
5. describe the algorithm that the server would use to identify peers in geographic proximity.
6. describe how you plan to implement privacy controls on location information.

Remember, a picture is worth a 1000 words, so try to provide pictures as well. Your document should be neatly typeset (hand written documents will not be accepted) and must be submitted by October 20th. Only one design document is required per team of students.

OpenMoko phones

You will be provided with an OpenMoko cell phone each to implement your project. You must contact Rick Crispin (Room 236, CoRE building) to collect your phones. You will be required to sign an agreement before you collect your phone. Borrowing this cell phone is akin to borrowing a library book, and you are expected to return these phones (including all accessories) to Rutgers at the end of the semester. Failing this, your degree will be placed on hold.

The Laboratory for Computer Science Research (LCSR) has set up a web page containing resources useful for the OpenMoko phones. The URL for this page is listed below and will constantly be updated throughout the semester in response to student queries.

<http://www.cs.rutgers.edu/resources/systems/openmoko>

Additionally, you can also send technical questions on the use of these cell phones and troubleshooting queries to help@cs.rutgers.edu with the phrase “**OpenMoko problem**” (without quotes) in the subject line of your email.

Project checkpoints and due dates

This project will proceed in several steps, and we will have checkpoints along the way to monitor your progress on the project. Checkpoints and their deadlines are listed below:

1. **Checkpoint:** Form a team of three or four students.

Deadline: October 1, 2008.

Teams consisting of fewer than three students or more than four students are not permitted without consent from the instructor.

2. **Checkpoint:** Sign agreement form and collect OpenMoko smart phones from the Computer Science department (point of contact: Rick Crispin).
Deadline: First week of October.
3. **Checkpoint:** Project design document.
Deadline: October 20, 2008.
The project design document is worth 20% of your grade on the project. Late submissions will not be accepted.
4. **Checkpoint:** Preliminary project demonstrations.
Deadline: Week after Thanksgiving.
By this time, you are expected to have most of your system working. That is, you should at least have the basic social networking application working without privacy controls. Project mentors will be assigned to each team and they will evaluate whether you have made sufficient progress on the project and will offer suggestions for improvement.
5. **Checkpoint:** Final project demonstrations.
Deadline: December 12, 2008.
This is the *final* project demonstration. You should have the entire system working and must be able to demonstrate all features of your system to the instructor. **You are also required to turn in all your source code by this date (failing which the project will be considered incomplete).**